FAKE NEWS DETECTION USING MACHINE LEARNING

A Mini Project Report

submitted by

HIBA NAFEESATH (MES20MCA-2020)

to

the APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree

of

Master of Computer Applications

Department of Computer Applications

MES College of Engineering
Kuttippuram, Malappuram - 679 582

February 2022

DECLARATION

I undersigned hereby declare that the project report **FAKE NEWS DETECTION USING MACHINE LEARNING**, submitted for partial fulfillment of the requirements for the award of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala, is a bonafide work done by me under supervision of Mr.Syed Feroze Ahamed M, Assistant Professor, Department of Computer Applications. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place:Kuttippuram

Date:

HIBA NAFEESATH (MES20MCA-2020)

DEPARTMENT OF COMPUTER APPLICATIONS
MES COLLEGE OF ENGINEERING, KUTTIPPURAM

CERTIFICATE

This is to certify that the report entitled **FAKE NEWS DETECTION USING MACHINE LEARNING** is a bonafide record of the Mini Project work carried out by **HIBA NAFEESATH (MES20MCA-2020)** submitted to the APJ Abdul Kalam Technological University, in partial fulfillment of the requirements for the award of the Master of Computer Applications, under my guidance and supervision. This report in any form has not been submitted to any other University or Institution for any purpose.

Internal Supervisor(s) External Supervisor(s)

Head Of The Department

Acknowledgements

At the very outset I would like to thank the almighty's mercy towards me over the years... I wish to express my sincere thanks to my project guide Ms.Priya J.D Assistant professor, Dept. of Computer Applications who guided me for the successful completeness of this project. I also thank her for valuable suggestions, guidance, constant encouragement, boundless corporation, constructive comments and motivation extended to me for completion of this project work.I would express my sincere thanks to my internal guide Mr.Syed Feroze Ahamed M Assistant professor for,immense guidance to complete the project successfully.I would like to express my sincere thanks to all the faculty members of Master of Computer Applications department for their support and valuable suggestion for doing the project work.Last but not least my graceful thanks to my parents, friends and also the persons who supported me directly and indirectly during the project.

HIBA NAFEESATH (MES20MCA-2020)

Abstract

News is the most vital source of information for common people about what is happening around the world. Newspapers are an authentic source of news, but nowadays social networks have become the emerging source of news. Due to easy access to these social networks, the news can be easily manipulated which gives rise to fake news. Fake news can be used for economic as well as political benefits. It can be used as a weapon to spread hate among the community which can harm society. So it is crucial to detect fake news to avoid its consequences. There is no existing platform that can verify the news and categorize it. This paper proposes a system that can be used for prediction of news to be real or fake. This system is based on natural language processing to extract features from the data and then these features are used for the training of machine learning classifiers such as Passive Aggressive Classifier(PAC).the classifier performance is evaluated on various parameters. Then the best performing classifier is deployed as a website using flask API for the prediction of the news

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

In traditional news making procedures, very limited and authorized individuals are involved and newspapers, radio,television were the only source of news. Due to these reasons news, credibility and authenticity are preserved. But in the era of internet, social network is becoming a news source of news. Easy and free access to these social networks makes the task of fabricating fake news and manipulating news a very effortless task. There is no authorize control point of these manipulated fake news which creates a question over there credibility and authenticity. The ease of getting direct news from the platform they mostly use has attracted the user. The reason to spread fake news can be social, political, and economical. Fake news in business can affect the stocks of the company leading to a huge capital loss. During the election campaign, fake news is used as a weapon against each other in a political war to defame the opposition. The most adverse effect is seen when it is used to spread communal hates which leads to riots. The Delhi riots are the best example of the destruction caused by fake news.Fake news about the COVID-19 in India lead to an attack on the medical team in various parts of the country and thus making the fight against the virus weak. The rate at which it spread is very fast due to which controlling the spread manually is not possible.There is no platform via which the user can check the credibility and authenticity of the news and where authorities can directly inform about the fake news prevailing. Due to which people can believe in the news which can be a trouble for them and as well for society also.In the existing system, the action is taken after the adverse impact had already hit society. The pro-

posed platform is useful for both common people and official authorities to prevent the spread of rumours in form of news.

1.1.1 Motivation

In our daily life Fake news is an important issue on social media. Using fake news the more criminal activity are happening in the world it causes defect on human life, to avoid and stop criminal activities by using these techniques Our work considers crowd signals for detecting fake news and is motivated by tools recently introduced by Facebook that enable users to flag fake news.

1.2 Objective

Our objective is to find the best Machine Learning Algorithms and Natural Learning Process methods for the prediction of news. Then the best performing model will be saved and will be linked to a user interface by which it can predict new input data. To achieve this objective training data have to go through various intermediate processes before giving it to Algorithms. The main parameters on which the performance of the model will be judge are accuracy of the model.

1.3 Contribution

In the current fake news corpus, there have been multiple instances where both supervised and unsupervised learning algorithms are used to classify text . However, most of the literature focuses on specific datasets or domains, most prominently the politics domain . Therefore, the algorithm trained works best on a particular type of article's domain and does not achieve optimal results when exposed to articles from other domains. Since articles from different domains have a unique textual structure, it is difficult to train a generic algorithm that works best on all particular news domains. In this paper, we propose a solution to the fake news detection problem using the machine learning ensemble approach. Our study explores different textual properties that could be used to distinguish fake contents and real contents. By using those properties, we train a machine learning algorithms using various ensemble methods.

The ensemble learners have proven to be useful in a wide variety of applications, as the learning models have the tendency to reduce error rate by using techniques such as bagging and boosting . These techniques facilitate the training of different machine learning algorithms in an effective and efficient manner. We also conducted extensive experiments on 4 real world publicly available datasets. The results validate the improved performance of our proposed technique using the commonly used performance metrics (accuracy)

1.4 Report Organization

The project report is divided into four sections. Section 2 describes literature survey. Section 3 describes the methodology used for implementing the project. Section 4 gives the results and discussions. Finally Section 5 gives the conclusion.

Chapter 2

Literature Survey

Kushal Agarwalla [1] proposed a system by comparing there ML classifier i.e. Logistic Regression, Naïve Bayes, Support Vector Machine using tokenizer as a feature. The maximum efficiency of system is achieved by using Support Vector Machine. The accuracy of model is only 81.25Chaitra K Hiramath [2] has compared both Machine Learning and Deep learning in the proposal. The maximum accuracy is given by Deep Neural Network which 91learning gave only 89Naïve Bayes, which is very basic classifier . A.Lakshmanarao [3] has used two NLP methods i.e. Bag of Words and TFIDF and compares the output of Machine Learning algorithms. The highest accuracy of 90.70by Random forest classifier. The results are shown by plotting confusion matrix. Abdullah-All-Tanvir [4] has compared both Machine Learning and Deep learning in the proposal. Count Vector, TFIDF, Word Embedding is used to build models. Highest efficiency of 89.34 From the above proposals, we can conclude that for higher accuracy of the model can be obtained by using a combination of the Natural Language Processing features. Further accuracy can be increased by smart cleaning and processing of data set. Words with meaning and importance should not be removed or manipulated.

Chapter 3

Methodology

3.1 Introduction

Our objective is to find the best Machine Learning Algorithms and Natural Learning Process methods for the prediction of news. Then the best performing model will be saved and will be linked to a user interface by which it can predict new input data . To achieve this objective training data have to go through various intermediate processes before giving it to Algorithms. The main parameters on which the performance of the model will be judge are accuracy of the model.

1. Data

Data is the prime ingredient of this project, as these data features are extracted using Natural Language Processing. By using these features of the data, Machine Learning Algorithms are trained and models are created. In this proposal, we have taken 6335 news with equal proportionality of fake and real. Data is saved in Comma Separated Value format. This data set is divided in the ratio of 80:20 for training and testing of algorithms.

2 .Data Cleaning

Data set is a chunk of data that is in raw form. Itmaycontaincertain symbols like digits, special characters, blank lines, and data without any label. These symbols should be removed as it is of no importance and it can also affect the performance of the model in an adverse manner.

3. Data Preprocessing

After data cleaning, data is now free of unwanted symbols. This data should be converted into the form which can be used for extracting the features easily. It includes the following

process:

•Converting each of the word in lowercase to avoid ambiguity between cases.

•Removal of the words which contain just one letter.

•Removal of the words that contain digits.

•Tokenize the data and removal of punctuations

•Removal of empty tokens.

•Stop Words: Stop words are useless words for NLP like “the”, “a”, “an”, “in”. This should be removed.

•Stemming:process of slicing the end or beginning of words with the intension of removing affixes.

4. Feature Selection

Feature Selection is the process where we select those features which contribute most to your prediction variable or output. Certain features are present in fake news, we have to extract them and accordingly, our classier is trained to predict the news. Here feature are the important words which appear in news. Natural language processing methods which are used to extract features are:

1)Bag of Words:

In this, each document is converted into a vector. First, all the words are taken out of the dataset forming bag of words. The vector of each document is the frequency of words present in it out of bag of word.

2) TF-IDF:

Bag of word depends only on the frequency of the words only. Certain words have very high frequency but are of no importance. To avoid this inverse document frequency is added which downscales words importance that appears a lot across documents. TF-IDF is word frequency scores that try to highlight more interesting words.

Here we use TF-IDF.

5.DEPLOYMENT

As CSM with TF-IDF came out to be the best method for the prediction, we need to deploy this model by making it platform independent and creating a user interface. Development of user interface and deployment of the model involves following steps:

1 .Pickling

Pickle is used for serializing and de-serializing Python object structures. Serialization refers to the process of converting an object in memory to a byte stream that can be stored on disk. The trained model is saved to disk using pickling by which we need not train the model every time we need it. We just need to deserialize it for predicting news input.

2. Pipeline

The data entered by the user will be in raw form. It should go through various intermediate processes like cleaning, processing, etc. before it can be used for prediction. Pipeline decides the flow of the data i.e. in what sequence data will go through various processes.

3. User Interface

An interface is created user to access the model without going into background detail of processing. It also removes the platform dependence i.e. it will only run on software on which it is modeled. In this project, a website constructed by HTML and CSS is used as a user interface.

4.API

API is the tool that is used to create an interface between the two applications. As our prediction model and user interface i.e. website are two applications that should be linked with each other to predict the input news and display it on the webpage.

This project was developed using Agile Development Model.The entire project was divided into four sprints.In the first sprint,Data collection and preprocessing is done. In second sprint,visualisation of data and split data into training and testing set,train the data.In third sprint,UI designing for register,login and checking form whether the news is fake or real. Fourth sprint is the generation of result.

3.2 Developing Environment

Hardware specification:

Processor : Intel Pentium Core i3 and above

Primary Memory : 4 GB RAM and above

Storage : 500 GB hard disk and above

Software specification:

Language :Python

Front end : Python Django

Back end : SQLite

Operating system : windows 7 and above

IDE : Visual Studio code,Jupyter Notebook

Others : HTML,CSS

Algorithm:Cosine Similarity TF-IDF vectorizer

Technique:Natural Language Processing

Data set:Fake news True news from Kaggle website

3.3 Natural Language Processing(NLP)

As human beings,when we read a sentense or a paragraph,we can interpret the words with the whole document and understand the context. Given today's volume of news ,it is possible to teach to a computer how to read and understand the differences between real news and the fake news using Natural Language Processing(NLP).

The building blocks are Data set and Machine Learning Algorithms.

3.4 Cosine Similarity Algorithm

Cosine similarity is a metric used to determine how similar the documents are irrespective of their size. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space. In this context, the two vectors I am talking about are arrays

containing the word counts of two documents. When plotted on a multi-dimensional space, where each dimension corresponds to a word in the document, the cosine similarity captures the orientation (the angle) of the documents and not the magnitude. If you want the magnitude, compute the Euclidean distance instead. The cosine similarity is advantageous because even if the two similar documents are far apart by the Euclidean distance because of the size they could still have a smaller angle between them. Smaller the angle, higher the similarity.

3.5 Passive Aggressive Classifier

The Passive-Aggressive algorithms are a family of Machine learning algorithms that aren't alright known by beginners and even intermediate Machine Learning enthusiasts. However, they will be very useful and efficient surely applications . Passive-Aggressive is considered algorithms that perform online learning.Their characteristics is that they remain passive when dealing with an outcome that has been correctly classified and become aggressive when a miscalculation takes place,thus constantly self-updating and adjusting.

3.6 Project Plan

A project plan that has a series of tasks laid out for the entire project, listing task duration, responsibility assignments, and dependencies. Plans are developed in this manner based on the assumption that the Project Manager, hopefully along with the team, can predict up front everything that will need to happen in the project, how long it will take, and who will be able to do it. Project plan is given below figure. The project has four sprints.

User Story ID	Task Name	Start Date	End Date	Days	Status
1	Sprint 1	27/12/2021	27/12/2021	2	Completed
2		28/12/2021	28/12/2021		Completed
3	Sprint 2	29/12/2021	29/12/2021	5	Completed
4		15/01/2022	16/01/2022		Completed
5		22/01/2022	22/01/2022		Completed
6		23/01/2022	23/01/2022		Completed
7	Sprint 3	26/01/2022	29/01/2022	3	Completed
8	Sprint 4	05/02/2022	06/02/2022	3	Completed
9		12/02/2022	12/02/2022		Completed

Figure 3.1: Project plan

3.7 User story

A key component of agile software development is putting people first, and user-stories put actual end users at the center of the conversation. Stories use non-technical language to provide context for the development team and their efforts. After reading a user story, the team knows why they are building what they're building and what value it creates. A user story is a tool used in agile software development to capture a description of a software feature from an end user perspective. The user story describes the type of user, what they want and why. A user story helps to create a simplified description of a requirement. User stories are one of the core components of an agile program. They help provide a user-focused framework for daily work which drives collaboration, creativity, and a better product overall. The user story of system is given in the below figure.

User Story ID	As a <type of user>	I want to <perform some task>	So that I can <achieve some goal>
1	User	Collection of Dataset	Fake news Dataset
2	User	Preprocessing of collected data(null values elimination)	Cleaned final dataset
3	User	Visualisation of input data	Graphical representation of data
4	User	Count no. of fake news & true news	Number of datas
5	User	Split data into training & testing set	80% -training data & 20%-testing
6	User	Train the data using TF-IDF vectorizer	Trained data
7	User	UI designing(a form to enter news for checking)	A form to enter news
8	User	Collect input from user & Testing	input from user & Tested data
9	User	generate output	Result(using Cosine similarity algorithm)

Figure 3.2: User story

3.8 Product backlog

A product backlog is a list of the new features, changes to existing features, bug fixes, infrastructure changes or other activities that a team may deliver in order to achieve a specific outcome.The product backlog is the single authoritative source for things that a team works on. That means that nothing gets done that isn't on the product backlog. Conversely, the presence of a product backlog item on a product backlog does not guarantee that it will be delivered. It represents an option the team has for delivering a specific outcome rather than a commitment.It should be cheap and fast to add a product backlog item to the product backlog, and it should be equally as easy to remove a product backlog item that does not result in direct progress to achieving the desired outcome or enable progress toward the outcome. The Scrum Product Backlog is simply a list of all things that needs to be done within the project. It replaces the traditional requirements specification artifacts. These items can have a technical nature or can be user-centric e.g. in the form of user stories.The product backlog of the system is given below figure.

User Story ID	Priority<High /Medium/Low >	Size(Hours)	Sprint	Status<Planned/Inprogress/ Completed>	Release Date	Release Goal
1	Medium	2	1	Completed	27/12/2021	Collection of datasets
2	High	3		Completed	28/12/2021	Preprocessing of collected data
3	Medium	3	2	Planned	29/12/2021	Visualisation of input data
4	High	2		Planned	15/01/2022-16/01/2022	Count no. of fake news & true news
5	Medium	5		Planned	22/01/2022	Split data into training & testing set
6	High	5		Planned	23/01/2022	Train the data
7	High	10	3	Planned	26/01/2022, 29/01/2022	UI designing(a form to enter news for checking)
8	High	20	4	Planned	05/02/2022-06/02/2022	Testing & collect input from user
9	High			Planned	12/02/2022	Generate Result

Figure 3.3: Product backlog

3.9 Sprint backlog

The sprint backlog is a list of tasks identified by the Scrum team to be completed during the Scrum sprint. During the sprint planning meeting, the team selects some number of product backlog items, usually in the form of user stories, and identifies the tasks necessary to complete each user story. Most teams also estimate how many hours each task will take someone on the team to complete.

Backlog item	Status & Completion date	Original Estimate in hours	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13
User Story#1,2		Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours
Data collection	27/12/2021	2	2	0	0	0	0	0	0	0	0	0	0	0	0
Preprocessing	28/12/2021	3	0	3	0	0	0	0	0	0	0	0	0	0	0
Userstory #3,4,5,6															
Visualisation & Training	23/01/2022	15	0	0	3	3	3	3	3	0	0	0	0	0	0
Userstory #7															
UI Designing	29/01/2022	10	0	0	0	0	0	0	0	4	4	2	0	0	0
User story #8,9															
Testing	12/02/2022	20	0	0	0	0	0	0	0	0	0	0	8	6	6
Total		50	2	3	3	3	3	3	3	4	4	2	8	6	6

Figure 3.4: Sprint backlog plan

3.10 Sprint Actual

Actual sprint backlog is what adequate sprint planning is actually done by project team there may or may not be difference in planned sprint backlog. The detailed sprint backlog (Actual) is given below.

Backlog item	Status & Completion date	Original Estimatein hours	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Completed<Y/N>
User Story#1,2		Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours
Data collection	27/12/2021	2	2	0	0	0	0	0	0	0	0	0	0	0	0	Y
Preprocessing	28/12/2021	3	0	3	0	0	0	0	0	0	0	0	0	0	0	Y
Userstory #3,4,5,6																
Visualisation & Training	23/01/2022	15	0	0	3	3	3	3	3	0	0	0	0	0	0	N
Userstory #7																
UI Designing	29/01/2022	10	0	0	0	0	0	0	0	4	4	2	0	0	0	N
User story #8,9																
Testing	12/02/2022	20	0	0	0	0	0	0	0	0	0	0	8	6	6	N
Total		50	2	3	3	3	3	3	3	4	4	2	8	6	6	N

Figure 3.5: Sprint1 actual

Backlog item	Status & Completion date	Original Estimatein hours	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Completed<Y/N>
User Story#1,2		Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours
Data collection	27/12/2021	2	2	0	0	0	0	0	0	0	0	0	0	0	0	Y
Preprocessing	28/12/2021	3	0	3	0	0	0	0	0	0	0	0	0	0	0	Y
Userstory #3,4,5,6																
Visualisation & Training	23/01/2022	15	0	0	3	3	3	3	3	0	0	0	0	0	0	Y
Userstory #7																
UI Designing	29/01/2022	10	0	0	0	0	0	0	0	4	4	2	0	0	0	N
User story #8,9																
Testing	12/02/2022	20	0	0	0	0	0	0	0	0	0	0	8	6	6	N
Total		50	2	3	3	3	3	3	3	4	4	2	8	6	6	N

Figure 3.6: Sprint2 actual

Backlog item	Status & Completion date	Original Estimatein hours	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Completed<Y/N>
User Story#1,2		Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours
Data collection	27/12/2021	2	2	0	0	0	0	0	0	0	0	0	0	0	0	Y
Preprocessing	28/12/2021	3	0	3	0	0	0	0	0	0	0	0	0	0	0	Y
Userstory #3,4,5,6																
Visualisation & Training	23/01/2022	15	0	0	3	3	3	3	3	0	0	0	0	0	0	Y
Userstory #7																
UI Designing	29/01/2022	10	0	0	0	0	0	0	0	4	4	2	0	0	0	Y
User story #8,9																
Testing	12/02/2022	20	0	0	0	0	0	0	0	0	0	0	8	6	6	N
Total		50	2	3	3	3	3	3	3	4	4	2	8	6	6	N

Figure 3.7: Sprint3 actual

Backlog item	Status & Completion date	Original Estimatein hours	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Completed<Y/N>
User Story#1,2		Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours
Data collection	27/12/2021	2	2	0	0	0	0	0	0	0	0	0	0	0	0	Y
Preprocessing	28/12/2021	3	0	3	0	0	0	0	0	0	0	0	0	0	0	Y
Userstory #3,4,5,6																
Visualisation & Training	23/01/2022	15	0	0	3	3	3	3	3	0	0	0	0	0	0	Y
Userstory #7																
UI Designing	29/01/2022	10	0	0	0	0	0	0	0	4	4	2	0	0	0	Y
User story #8,9																
Testing	12/02/2022	20	0	0	0	0	0	0	0	0	0	0	8	6	6	Y
Total		50	2	3	3	3	3	3	3	4	4	2	8	6	6	Y

Figure 3.8: Sprint4 actual

Chapter 4

Results and Discussions

4.1 Datasets

In this project dataset used is fakenews dataset and truenews dataset which is downloaded from kaggle website. Fakenews dataset contains a lots of fakenews data and truenews dataset contains lots of truenews.Divide dataset into training data and test data.we train and create a model using training data from the dataset and we test using test data to detect whether the news is fake or not.

4.2 Results

Best Model i.e. combination of Machine learning algorithms with NLP method is selected after comparing the results of the models. These models are made by taking ML algorithms with a combination of NLP methods. From the comparison of the parameters, we can conclude that the Passive Aggressive classifier with TF-IDF method is the best performer in terms of accuracy 93 percentage. The model with the best performance is obtained by tuning the parameter of pac architecture. we can conclude that the model can predict both true and false input with equal precision and recall. This reduces the error in the prediction by the model.

Screeshots

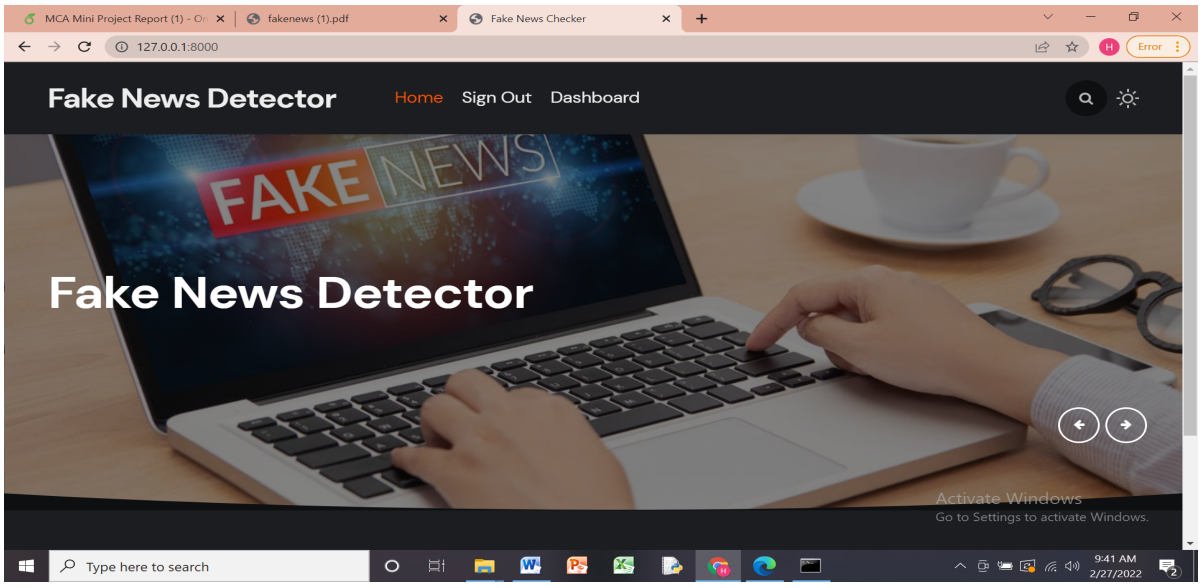

Figure 4.1: Home page

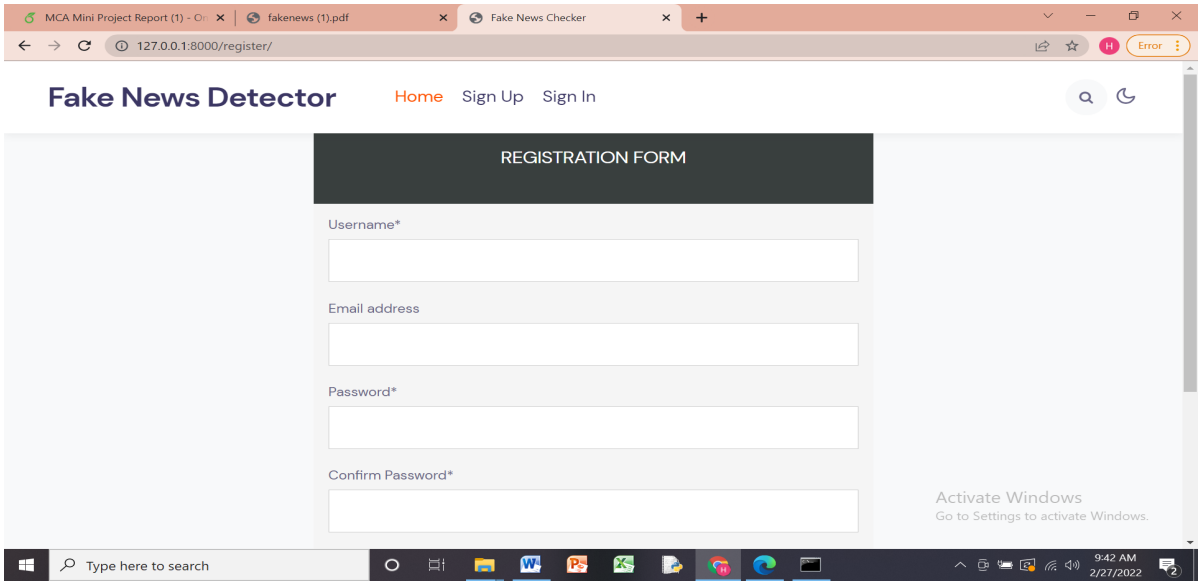

Figure 4.2: Registration

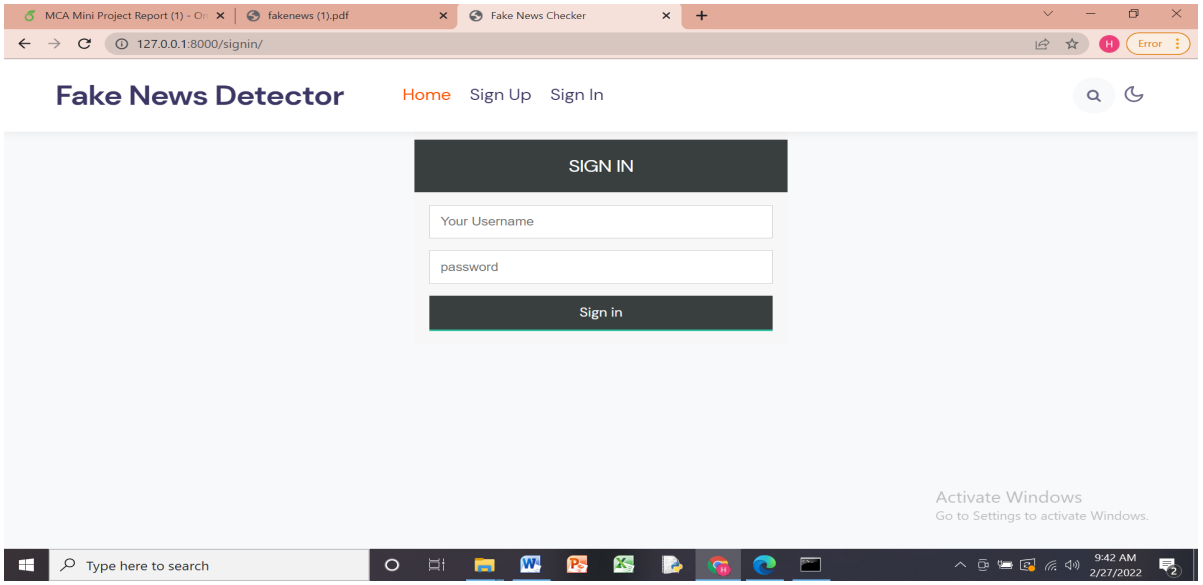

Figure 4.3: Log in

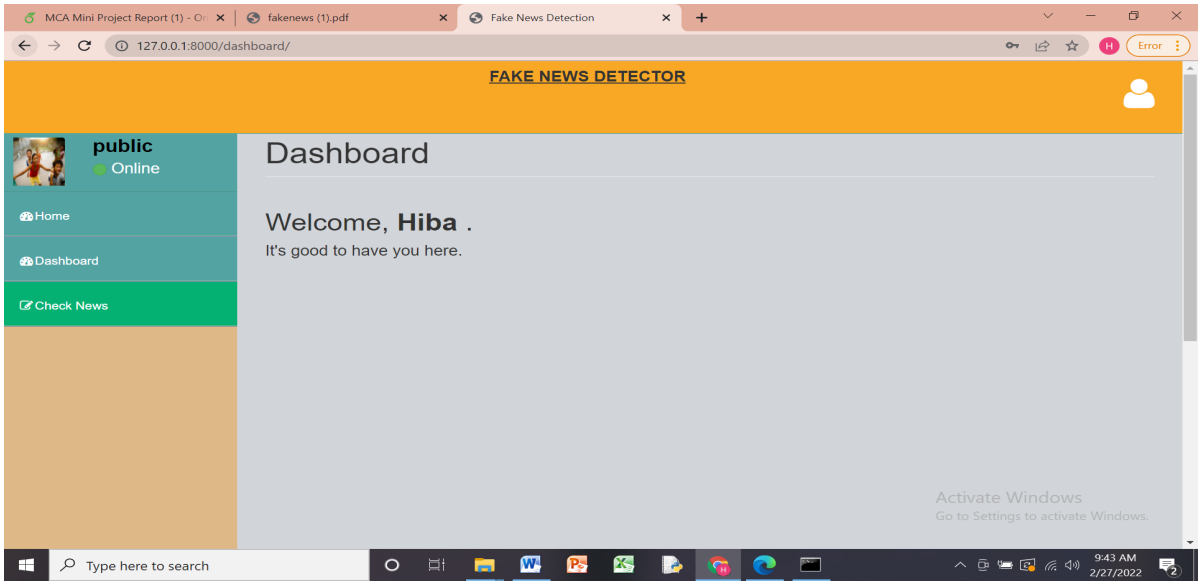

Figure 4.4: Loged Home

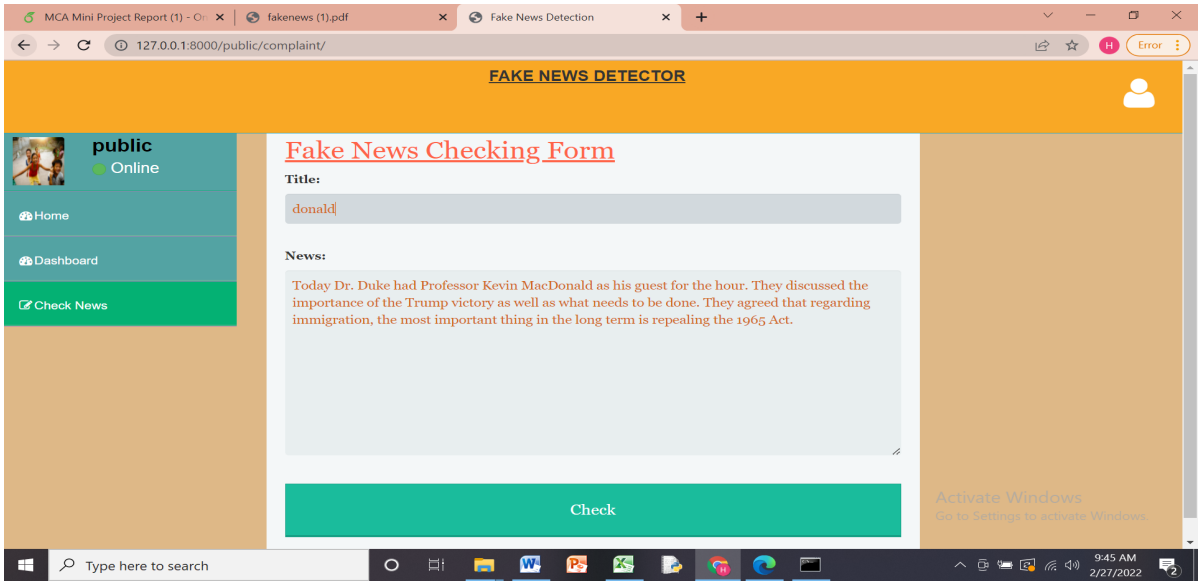

Figure 4.5: News Checking

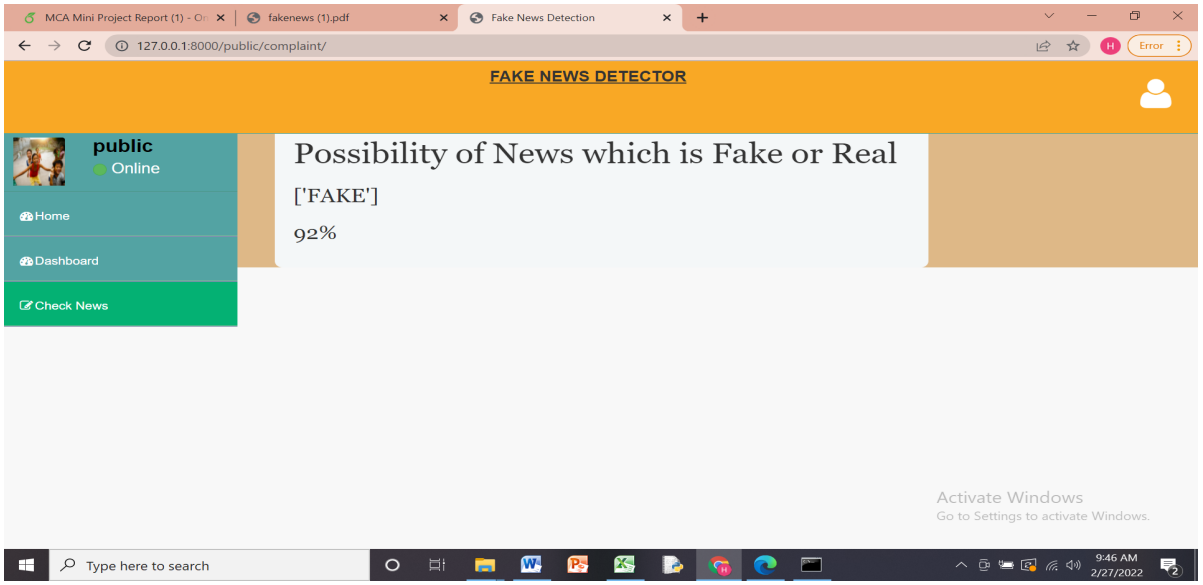

Figure 4.6: Result

Chapter 5

Conclusions

A platform i.e. website is created which is linked to a trained Machine Learning model. Cosine similarity with TF-IDF NLP method is trained and use for the prediction of new news input by the user. This trained model link to the platform is capable of predicting news to be fake or real with an accuracy of 93In this platform, users can enter the news and it will predict whether the news is real or fake. It will show output as real or fake with the possibility of news to be real or fake.

References

[1] **Kushal Agarwalla, Shubham Nandan, Varun Anil Nair, D. Deva Hema, "Fake News Detection using Machine Learning and Natural Language Processing," International Journal of Recent Technology and Engineering (IJRTE), Volume-7, Issue-6, March 2019.**.

[2] **Chaitra K Hiramath, Prof. G.C Deshpande, Fake News Detection Using Deep Learning Techniques, 1st International Conference on Advances in Information Technology,2019.**

[3] **A.Lakshmanarao, Y.Swathi, T. Srinivasa Ravi Kiran, "An Effecient Fake News Detection System Using Machine Learning," International Journal of Innovative Technology and Exploring Engineering (IJITEE), Volume-8, Issue-10, August- 2019.**

[4] **Abdullah-All-Tanvir, Ehesas Mia Mahir, Saima Akhter, Mohammad Rezwanul Huq, "Detecting Fake News using Machine Learning and Deep Learning Algorithms," 7th International Conference on Smart Computing Communications (ICSCC),2019.**

Appendix

Source Code

```
 ####training model code####
#code to mount the google colab with google drive
#from google.colab import drive
#drive.mount('/content/drive')
   ########################################

   import os

path='C:\\Users\\windos\\Desktop\\Project\\mysite\\'
#path='/content/drive/MyDrive/News'
!ls

os.chdir(path)

   ########################################

import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import PassiveAggressiveClassifier
from sklearn.metrics import accuracy_score, confusion_matrix

#Read the Data
df = pd.read_csv(path+'news.csv',index_col=False)
df=df.drop('Unnamed: 0',1)
df.head()
df.shape
labels = df['label']
labels.head()

#split into train test
x_train, x_test, y_train, y_test = train_test_split(df['text'], labels, test_size = 0.2, random_state = 7)

import pickle
#Initialize a TfidfVectorizer
tfidf_vectorizer = TfidfVectorizer(stop_words='english', max_df = 0.7)

print(type(x_train))
print(x_train)

#Fit and transform train and test set
tfidf_train = tfidf_vectorizer.fit_transform(x_train)
tfidf_test = tfidf_vectorizer.transform(x_test)
```

Appendix

```
# Save the vectorizer
vec_file = 'C:\\Users\\windos\\Desktop\\Project\\mysite\\vectorizer.pickle'
pickle.dump(tfidf_vectorizer, open(vec_file, 'wb'))

# Initialize a PassiveAggressiveClassifier and fit the train data
pac = PassiveAggressiveClassifier(max_iter = 50)
pac.fit(tfidf_train, y_train)

#Predict on the test set and calculate accuracy
y_pred = pac.predict(tfidf_test)
score = accuracy_score(y_test, y_pred)
print(f'Accuracy: {round(score * 100, 2)}')


#testing

sample_text=["The Battle of New York: Why This Primary Matters"]

loaded_vectorizer = pickle.load(open('vectorizer.pickle', 'rb'))
output_label=pac.predict(loaded_vectorizer.transform(sample_text))
print(output_label)

#!pip install pandas
#!pip install numpy
#!pip install wordcloud
import pandas as pd
import numpy as np

truenews = pd.read_csv('C:\\Users\\windos\\Desktop\\fakenews\\true.csv')
fakenews = pd.read_csv('C:\\Users\\windos\\Desktop\\fakenews\\fake.csv')

fakenews.head()
truenews.head()
fakenews.describe()
truenews.describe()

#Data Cleaning
from nltk.corpus import stopwords
import string
def process_text(s):

# Check string to see if they are a punctuation
   nopunc = [char for char in s if char not in string.punctuation]

# Join the characters again to form the string.
   nopunc = ''.join(nopunc)

# Convert string to lowercase and remove stopwords
   clean_string = [word for word in nopunc.split() if word.lower() not in stopwords.words('english')]
   return clean_string

# Tokenize the text :Convert the normal text strings in to a list of tokens (words that we actually want)
#rerun, takes LOOOONG
truenews['True/Fake']='True'
fakenews['True/Fake']='Fake'
# Combine the 2 DataFrames into a single data frame
news = pd.concat([truenews, fakenews])
news["Article"] = news["title"] + news["text"]
news.sample(frac = 1) #Shuffle 100%

news['Clean Text'] = news['Article']
print(news['Clean Text'])
```

Appendix

```
news.describe()
import nltk
nltk.download('punkt')

from nltk.stem.porter import PorterStemmer

stem_words = PorterStemmer()
#!pip install wordcloud

def stemming(data):
   stemmed_content = re.sub('[^a-zA-Z]',' ',data)
   stemmed_content = stemmed_content.lower()
   stemmed_content = stemmed_content.split()
   stemmed_content = [stem_words.stem(word) for word in stemmed_content if not word in stopwords.words('english')]
   stemmed_content = ' '.join(stemmed_content)
   return stemmed_content

#!pip install wordcloud
from wordcloud import WordCloud, STOPWORDS
from PIL import Image
from nltk.corpus import stopwords
nltk.download('stopwords')
stopwords = set(STOPWORDS)
mask=np.array(Image.open("cloud.png"))
wordcloud = WordCloud(
                  background_color='black',
                  stopwords=stopwords,
                  max_words=3000,
                  random_state=42)
wordcloud.generate(str(news["Article"]))
wordcloud.to_file("wc.png")
path="wc.png"
display(Image.open(path))
print(wordcloud)

#!pip install wordcloud
from wordcloud import WordCloud, STOPWORDS
from PIL import Image
from nltk.corpus import stopwords
nltk.download('stopwords')
stopwords = set(STOPWORDS)
mask=np.array(Image.open("cloud.png"))
wordcloud = WordCloud(
                  background_color='black',
                  stopwords=stopwords,
                  max_words=3000,
                  random_state=42)
wordcloud.generate(str(truenews))
wordcloud.to_file("wc.png")
path="wc.png"
display(Image.open(path))
print(wordcloud)

#!pip install wordcloud
from wordcloud import WordCloud, STOPWORDS
from PIL import Image
from nltk.corpus import stopwords
nltk.download('stopwords')
stopwords = set(STOPWORDS)
mask=np.array(Image.open("cloud.png"))
wordcloud = WordCloud(
                  background_color='black',
                  stopwords=stopwords,
```

Appendix

```
                  max_words=3000,
                  random_state=42)
wordcloud.generate(str(fakenews))
wordcloud.to_file("wc.png")
path="wc.png"
display(Image.open(path))
print(wordcloud)
```